A Two Parameter Exact Penalty Function for Nonlinear Programming

by

I.D. Coope and C.J. Price
Department of Mathematics and Statistics,
University of Canterbury, Christchurch, New Zealand

No. 96 July, 1993

Abstract – A sequential quadratic programming algorithm for nonlinear programmes using an ℓ_∞ exact penalty function is described. Numerical results are also presented. These results show that the algorithm is competitive with other exact penalty function based algorithms, and that the inclusion of the second penalty parameter can be advantageous.

Keywords: Nonlinear Optimization, Exact Penalty Functions.

1 Introduction

Nonlinear Programming problems (NLP) arise in many practical situations, and algorithms based on exact penalty functions have proved particularly effective in solving such problems.

A common approach which yields global convergence is the use of Sequential Quadratic Programming (SQP) techniques in conjunction with an exact penalty function. This approach was introduced and popularised for the ℓ_1 penalty function in the papers of Han [3], and Powell [10]. In this paper, however, attention is directed to an ℓ_∞ Exact Penalty Function (EPF) following the approach taken in [8, 7]. The purpose of this paper is to show that there are some advantages to be gained from using a two parameter exact penalty function based on the infinity norm of constraint violations.

The NLP considered is of the form:

$$\min_{x \in R^n} f(x) \text{ subject to } c(x) \leq 0, \tag{1}$$

where the objective function f, mapping R^n into R, and the constraint function c, mapping R^n into R^m, are continuously differentiable. For convenience, discussion of the problem has been restricted to inequality constraints, but nonlinear equality constraints and simple bounds, together with other linear constraints may be present in a more general formulation.

Under an appropriate constraint qualification (for example [2]) the Karush-Kuhn-Tucker (KKT) conditions are necessary for optimality, and these are assumed to hold at all solutions to the NLP problem (1) which are of interest.

Assumption 1.1 *Let x^* be any optimal point of the NLP (1). Then there exists a* *vector of Lagrange multipliers* λ^* *in* R^m *such that*

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla c_i(x^*) = 0 \tag{2}$$

with

$$\lambda_i^* c_i(x^*) = 0, \;\; \lambda_i^* \geq 0, \;\; c_i(x^*) \leq 0, \;\; \forall i = 1, \ldots, m. \tag{3}$$

2 The Penalty Function Problem

The approach taken in this paper is to replace the NLP by the more tractable problem of minimizing a non-differentiable penalty function chosen so that solutions of the NLP are also solutions of the Penalty Function Problem (PFP). The exact penalty function used in this paper is based on the infinity norm of the constraint violations

$$\theta(x) = \max_{1 \leq i \leq m} [c_i(x)]_+ \tag{4}$$

and has the form

$$\Phi(x) = f(x) + \mu\theta(x) + \tfrac{1}{2}\nu\theta^2(x). \tag{5}$$

The penalty parameters μ and ν are restricted to $\mu > 0$ and $\nu \geq 0$. The penalty function Φ may be viewed as a hybridization of the single parameter exact penalty function $f + \mu\theta$ and the classical quadratic penalty function, and is referred to as the hybrid (or the two parameter) exact penalty function.

Clearly $\theta(x)$ (and hence Φ) is continuous $\forall x \in R^n$. However $\theta(x)$ is not usually differentiable at points where $c_i(x) = \theta(x)$ for more than one value of i, or where $c_i(x) = \theta(x) = 0$ for some i. Nevertheless the directional derivative

$$D_p\theta(x) = \lim_{\alpha \to 0^+} [\theta(x + \alpha p) - \theta(x)] / \alpha \tag{6}$$

exists for all $x, p \in R^n$.

Lemma 2.1 *If $\{c_i(x)\}_{i=1}^m$ are continuously differentiable functions from R^n to R and if $\theta(x)$ is defined by (4), then for any direction $p \in R^n$, the directional derivative $D_p\theta(x)$ exists and*

$$D_p\theta(x) = \begin{cases} \max_{i\in I(x)} p^T \nabla c_i(x) & \text{if } \theta(x) > 0 \text{ and } I(x) \neq \emptyset \\ \max_{i\in I(x)} [p^T \nabla c_i(x)]_+ & \text{if } \theta(x) = 0 \text{ and } I(x) \neq \emptyset \\ 0 & I(x) = \emptyset \end{cases} \tag{7}$$

where $I(x) = \{i : c_i(x) = \theta(x)\}$.

Definition 2.2 *For fixed values of μ and ν, a point $\overline{x}$ is a* **critical point** *of $\Phi(x)$ iff for all $p \in R^n$ the directional derivative $D_p\Phi(\overline{x})$ is non-negative.*

The solution set of the PFP with fixed values for μ and ν is defined as the set of critical points of $\Phi(x)$.

If solving the PFP is to yield a solution of the NLP, it is highly desirable that the PFP's solution set be contained in (and ideally equal to) the NLP's solution set. This can be achieved to a limited extent by a suitable choice of μ, for any ν as the following theorem shows.

Theorem 2.3 *Let x^* be an optimal solution of the NLP (1) at which conditions (2,3) hold, and let λ^* be a vector of Lagrange multipliers satisfying these conditions for which $\|\lambda^*\|_1$ is minimal. If μ satisfies*

$$\mu > \|\lambda^*\|_1 \tag{8}$$

then x^* is a critical point of $\Phi(x)$.

Conversely, if x^* is both feasible and a critical point of Φ for some $\mu > 0$ and $\nu \geq 0$, then x^* is a KKT point of the NLP.

Proof: Firstly, if x^* is both feasible and satisfies (2) and (3), then $\theta(x^*) = 0$, and $\lambda_i^* = 0 \; \forall i \notin I(x^*)$. Hence, on using $c_i^* = c_i(x^*)$, for any p:

$$\begin{aligned}
D_p\Phi(x^*) &= p^T\nabla f(x^*) + (\mu + \nu\theta(x^*))D_p\theta(x^*) \\
&= -\sum_{i\in I(x^*)} \lambda_i^* p^T \nabla c_i^* + \mu \max_{j\in I(x^*)} \left[p^T\nabla c_j^*\right]_+ \\
&\geq -\sum_{i\in I(x^*)} \lambda_i^* \max_{j\in I(x^*)} \left[p^T\nabla c_j^*\right]_+ + \mu \max_{j\in I(x^*)} \left[p^T\nabla c_j^*\right]_+ \\
&= (\mu - \|\lambda^*\|_1) \max_{i\in I(x^*)} \left[p^T\nabla c_i^*\right]_+ ,
\end{aligned}$$

which shows that $D_p\Phi(x^*) \geq 0$ holds if $\mu \geq \|\lambda^*\|_1$, and hence that x^* is a critical point of Φ.

Secondly, if x^* is a critical point of Φ for some μ and ν, then

$$\forall x \text{ near } x^*, \quad \Phi(x) \geq \Phi(x^*) + D_{x-x^*}\Phi(x^*) + o(\|x - x^*\|) \geq \Phi(x^*) + o(\|x - x^*\|).$$

Now $\Phi \equiv f$ on the NLP's feasible region, and so x^* is a KKT point of the NLP. □

3 Calculating Descent Directions for $\Phi(x)$.

It has been shown in the previous section that, under the appropriate assumptions, the NLP may be replaced by the problem of finding feasible local minimizers of the PFP. The PFP is minimized by an iterative method which chooses a direction of descent at each iteration before applying a line search to reduce Φ. In order to determine a suitable descent direction at the k^{th} iterate $x^{(k)}$, a continuous piecewise quadratic approximation $\Psi^{(k)}(p)$ to Φ near $x^{(k)}$ is defined as follows:

$$\Psi^{(k)}(p) = f(x^{(k)}) + p^T\nabla f(x^{(k)}) + \tfrac{1}{2}p^T H^{(k)} p + \mu^{(k)}\zeta(p) + \tfrac{1}{2}\nu^{(k)}\zeta^2(p), \tag{9}$$

$$\text{where } \; \zeta(p) = \max_{i=1,\ldots,m} \left[c_i(x^{(k)}) + p^T\nabla c_i(x^{(k)})\right]_+ , \tag{10}$$

and where $H^{(k)}$ is positive definite. Using Taylor series expansions,

$$\Phi(\mu^{(k)}, \nu^{(k)}; p) = \Psi^{(k)}(p) + o(\|p\|) \quad \text{for } p \text{ small.} \tag{11}$$

Clearly Ψ is strictly convex in p, and the level set $\{p \in R^n : \Psi(p) \leq \Psi(0)\}$ is bounded for all $\mu > 0$ and all $\nu \geq 0$. Thus $\Psi^{(k)}$ has a unique global minimizer $p^{(k)}$. Because $p^{(k)}$ also solves the quadratic programming problem

$$\min_{p,\zeta} \left\{ p^T \nabla f^{(k)} + \tfrac{1}{2} p^T H^{(k)} p + \mu^{(k)} \zeta + \tfrac{1}{2} \nu^{(k)} \zeta^2 \right\} \tag{12}$$

$$\text{subject to } \; c_i^{(k)} + p^T \nabla c_i^{(k)} \leq \zeta \;\; \forall i = 1, \ldots, m, \;\; \text{and} \;\; \zeta \geq 0, \tag{13}$$

both problems are referred to as $\ell_\infty \mathrm{QP}^{(k)}$.

Theorem 3.1 *Let $f^{(k)}$, $c^{(k)}$ and $\theta^{(k)}$ denote $f(x^{(k)})$, $c(x^{(k)})$, and $\theta(x^{(k)})$ respectively, and let $p = p^{(k)}$ and $\zeta = \zeta^{(k)}$ denote the unique solution to $\ell_\infty QP^{(k)}$, where $H^{(k)}$ is positive definite. Further, let $\lambda^{(k)}$ denote an optimal Lagrange multiplier vector (which need not be unique) for this $\ell_\infty QP$ for which $\|\lambda^{(k)}\|_1$ is least. If $p^{(k)} \neq 0$,*

$$\zeta^{(k)} \leq \theta^{(k)} \tag{14}$$

and

$$\mu + \nu \theta^{(k)} \geq \|\lambda^{(k)}\|_1 \tag{15}$$

then $p^{(k)}$ is a descent direction for $\Phi(x)$ at $x^{(k)}$.

Proof: The KKT conditions (2), (3) for $\ell_\infty \mathrm{QP}^{(k)}$ are

$$[Hp + \nabla f + \sum_{i=1}^{m} \lambda_i \nabla c_i]^{(k)} = 0 \tag{16}$$

$$[\lambda_i (p^T \nabla c_i + c_i - \zeta)]^{(k)} = 0, \text{ and } \lambda_i^{(k)} \geq 0, \;\; \forall i = 1, \ldots, m, \tag{17}$$

$$\zeta \lambda_\zeta = 0, \;\; \lambda_\zeta \leq 0, \text{ and } \mu + \nu\zeta = \|\lambda\|_1 + \lambda_\zeta. \tag{18}$$

Here λ_ζ is the Lagrange multiplier for the constraint $\zeta \geq 0$. Therefore, using (16)

$$\begin{aligned} D_{p^{(k)}}\Phi(x^{(k)}) &= (p^T\nabla f)^{(k)} + (\mu + \nu\theta^{(k)})D_{p^{(k)}}\theta(x^{(k)}) \qquad (19) \\ &= -(p^T Hp)^{(k)} - \sum_{i=1}^{m}(\lambda_i p^T \nabla c_i)^{(k)} + (\mu + \nu\theta^{(k)})D_{p^{(k)}}\theta(x^{(k)}) \\ &= -(p^T Hp)^{(k)} - \sum_{i=1}^{m}\lambda_i^{(k)}(\zeta^{(k)} - c_i^{(k)}) + (\mu + \nu\theta^{(k)})D_{p^{(k)}}\theta(x^{(k)}). \end{aligned}$$

Clearly $D_{p^{(k)}}\theta(x^{(k)}) = D_{p^{(k)}}\zeta(0)$ from (10). Now $D_{p^{(k)}}\zeta(0) \leq \zeta(p^{(k)}) - \zeta(0)$ because $\zeta(p)$ is convex. Also $\zeta^{(k)} = \zeta(p^{(k)}) \leq \zeta(0) = \theta^{(k)}$, and so $D_{p^{(k)}}\theta(x^{(k)}) \leq \zeta^{(k)} - \theta^{(k)}$. Applying this result to (19) gives the inequality

$$D_{p^{(k)}}\Phi(x^{(k)}) \leq -(p^T Hp)^{(k)} - \left(\mu + \nu\theta^{(k)} - \|\lambda^{(k)}\|_1\right)(\theta^{(k)} - \zeta^{(k)}). \qquad (20)$$

Clearly $p^{(k)}$ is a descent direction under the stated conditions. $\square$

Corollary 3.2 *By comparing conditions (16, 17) with conditions (2, 3) it can be seen that, if $p^{(k)} = 0$ solves the $\ell_\infty QP$, then $x^{(k)}$ is a KKT point for the problem*

$$\min_{x \in R^n} f(x) \ \ \textit{subject to} \ \ c(x) \leq \theta(x^{(k)}).$$

Additionally, if $\zeta^{(k)} \leq \theta^{(k)}$ (respectively $<$), then by the convexity of $\zeta(p)$, if follows that $D_{p^{(k)}}\zeta(0) \leq 0$ (respectively $<$). Equation (10) implies $p^{(k)}$ is a direction of non-ascent (respectively strict descent) for θ at $x^{(k)}$.

Corollary 3.3 *By comparing conditions (2,3) with conditions (16,17) it can be seen that if $p^{(k)}, \zeta^{(k)} = 0$ solves $\ell_\infty QP^{(k)}$, then $x^{(k)}$ solves (1).*

4 An ℓ_∞ Exact Penalty Function Algorithm

The results of the preceding sections are used to define an effective algorithm for NLP problems. For purposes of ensuring convergence, the following bound is imposed on

$\ell_\infty \mathrm{QP}^{(k)}$ at each iteration

$$\|p^{(k)}\|_\infty \leq S_{\mathrm{bound}}. \tag{21}$$

The following algorithm was used to test the effectiveness of including the second penalty parameter. Many other algorithms based on the hybrid penalty function are of course possible.

Algorithm A.

1. INITIALIZATION.

$$\mu^{(1)} = 1, \quad \nu^{(1)} = 1, \quad k = 1, \quad H^{(1)} = I_n,$$

$$\rho = 0.02, \quad S_{\mathrm{bound}} = 10^{10}, \quad \theta_{\mathrm{crossover}} = 1, \quad \theta_{\mathrm{cap}} = 100,$$

$$\kappa_1 = 1.2, \quad \kappa_2 = 1.5, \quad \kappa_3 = 1.2, \quad \kappa_4 = 4.$$

2. UPDATE H, AND THE PENALTY PARAMETERS. This step is omitted from the first iteration. H is updated using the BFGS update provided this maintains positive definiteness; otherwise H is not updated. The penalty parameters are updated as follows:

$$\text{if } \theta^{(k)} \leq \theta_{\mathrm{crossover}} \text{ and } \mu^{(k)} < \kappa_1 \|\lambda^{(k)}\|_1$$

$$\text{then set } \mu^{(k+1)} = \kappa_2 \|\lambda^{(k)}\|_1 \text{ and } \nu^{(k+1)} = \nu^{(k)}.$$

$$\text{If } \theta^{(k)} > \theta_{\mathrm{crossover}} \text{ and } \mu^{(k)} + \nu^{(k)}\theta^{(k)} < \kappa_1 \|\lambda^{(k)}\|_1$$

$$\text{then set } \nu^{(k+1)} = \frac{\kappa_2 \|\lambda^{(k)}\|_1 - \mu^{(k)}}{\theta^{(k)}} \text{ and } \mu^{(k+1)} = \mu^{(k)}.$$

Otherwise the penalty parameters are not altered.

3. SOLVE THE ℓ_∞QP. If $\theta^{(k)} > \theta_{\mathrm{cap}}$ then the capping constraint $\zeta \leq \theta^{(k)}$ is also imposed on the ℓ_∞QP (12,13). This ℓ_∞QP is then solved. If the capping

constraint is not active at the ℓ_∞QP's solution then the algorithm proceeds directly to step 4. Otherwise the penalty parameters are updated as described in step 2, except that $\|\lambda^{(k)}\|_1$ is replaced by $\mu^{(k)} + \nu^{(k)}\theta^{(k)} + |\xi|$, where ξ is the Lagrange multiplier of the capping constraint. The ℓ_∞QP (12,13) is then re-solved.

4. ATTEMPT THE PROPOSED STEP. If both of the following conditions hold:

$$\text{first } \Phi(x^{(k)}) - \Phi(x^{(k)} + p^{(k)}) \geq \rho \left[\Psi^{(k)}(0) - \Psi^{(k)}(p^{(k)})\right],$$

and second, either the penalty parameters were not altered in step 2 or the inequality $\theta(x^{(k)} + p^{(k)}) \leq \theta(x^{(k)})$ is satisfied, then the proposed step $p^{(k)}$ is accepted and the algorithm proceeds to step 7. Otherwise execution continues at the next step.

5. CALCULATE THE MARATOS EFFECT CORRECTION VECTOR. Solve the following QP for the second order correction $t^{(k)}$:

$$\min_{t \in R^n} \|t\|_2^2,$$

$$\text{such that } t^T \nabla c_i(x^{(k)}) + c_i(x^{(k)} + p^{(k)}) \geq 0 \ \ \forall i \in T,$$

where T is the set of indices of the constraints active at the QP's solution in step 2. If $\|t^{(k)}\|_2 \geq \|p^{(k)}\|_2$, then set $t^{(k)} = 0$. This vector is essentially that of Mayne and Polak [5].

6. DO THE ARC SEARCH. Consider successive values of the sequence $1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots$ as trial values of α. If $t^{(k)} = 0$ then omit the first member of the sequence, otherwise start with $\alpha = 1$. Accept the first trial value of α which satisfies the following two conditions: first

$$\Phi(x^{(k)}) - \Phi(x^{(k)} + q^{(k)}(\alpha)) \geq \rho\alpha \left[\Psi^{(k)}(0) - \Psi^{(k)}(p^{(k)})\right], \tag{22}$$

and second, if the penalty parameters were altered in step 3, then the step $q^{(k)}(\alpha)$ is also required to satisfy the condition $\theta(x^{(k)} + q^{(k)}(\alpha)) \leq \theta^{(k)}$. After a satisfactory value of α has been found, set $x^{(k+1)} = x^{(k)} + q^{(k)}(\alpha)$.

7. CHECK THE STOPPING CONDITIONS. The algorithm halts if either the length of the previous step $\|x^{(k)} - x^{(k-1)}\|$ is less than 10^{-8}, or both of the following conditions hold:

$$\theta^{(k)} < 10^{-5} \quad \text{and} \quad \left\| \nabla f(x^{(k)}) + \sum_{i \in \mathcal{A}^{(k)}} \lambda_i^{(k)} \nabla c_i(x^{(k)}) \right\|_2 < 10^{-5},$$

where $\mathcal{A}^{(k)} = \{i \in 1, \ldots, m : |c_i(x^{(k)})| < 10^{-5}\}$. Otherwise k is incremented, and the algorithm proceeds to step 2.

The convergence properties of algorithm A are studied in [9] and [7], and are summarised in the following theorem.

Theorem 4.1 *Given:*

1. *The sequence of iterates $\{x^{(k)}\}$ generated by the algorithm is bounded.*

2. *The sequence of matrices $\{H^{(k)}\}$ is bounded in norm.*

3. *The parameters μ and ν are altered only a finite number of times.*

Then every cluster point of the sequence of iterates $\{x^{(k)}\}$ generated by the algorithm is a critical point of $\Phi(\mu, \nu, x)$, where μ and ν are at their final values.

Proof. A proof of this theorem is given in [9]. Alternatively, the NLP (1) may be viewed as a specialization of the semi-infinite programming problem described in [7]. The specialization of the exact penalty function used in [7] is equivalent to (5), and so a convergence proof for algorithm A may be constructed along the lines of the convergence proof given in [7]. The proof presented in [9] exploits the finiteness of the NLP's constraint set, and hence is simpler. □

5 Numerical Results.

The algorithm was tested on a variety of problems, and results are listed in Tables 1 to 4. The legend for these tables is as follows: j denotes the number of times the objective and constraint functions are evaluated; h is the number of objective and constraint gradient evaluations performed (which equals the number of iterations performed); and r denotes the magnitude of the KKT residual calculated as described in step 7 of the algorithm. The subscripts S, P, B, and C respectively refer to results given by Sahba [12], by Powell [10], by Bartholomew-Biggs [1], and those generated by algorithm A. The superscript $\sharp$ refers to the value a quantity takes at the final iterate generated by the algorithm.

The algorithm was tested on those problems listed in [4] for which results are given in [1]. These problems are tagged using the numbers given to them in [4], and the results are listed in Table 1. For problem 46 the initial values $\mu^{(0)} = \nu^{(0)} = 1$ were excessive; the results listed in Table 1 for this problem were generated using $\mu^{(0)} = \nu^{(0)} = 10^{-4}$. The algorithm solved problem 46 without difficulty using $\mu^{(0)} = \nu^{(0)} = 1$; 435 function evaluations and 99 gradient evaluations were performed in doing so. Problem 46 is interesting in that the Hessian of the Lagrangian is singular at the solution.

The algorithm was also tested on problems 1, 2, 3, 4, 6, and 7 of the Colville set (hereafter C1, C2 etc). Problems C2 and C3 have an 'F' or 'I' postfix to indicate whether the feasible or infeasible initial point was used. The results for these problems are listed in Table 2. The last three rows of Table 2 list the results on the

problem	j_B	h_B	j_C	h_C	$\theta^\sharp$	$r^\sharp$	$\mu^\sharp$	$\nu^\sharp$
7	10	9	10	7	2.3e-7	1.4e-6	1	1
27	20	18	26	22	2.9e-12	5.4e-6	1	1
39	14	14	14	13	1.7e-9	9.3e-6	2.77	1
46	34	24	20	14	9.6e-11	3.7e-6	0.031	1E-4
52	11	9	13	8	2.8e-17	5.2e-6	18.56	4.97
56	20	16	13	9	3.4e-9	1.9e-6	2.839	1
78	10	10	10	7	7.0e-12	8.0e-7	3.15	1

Table 1: Results for the problems solved by Bartholomew-Biggs.

Wright 9 problem. Row W9a uses the starting point $x_a^{(0)} = (1,1,1,1,1)^T$, rows W9b and W9c use the initial point $x_{bc}^{(0)} = (1.091, -3.174, 1.214, -1.614, 2.134)^T$. The initial values $\mu^{(0)} = \nu^{(0)} = 1$ were insufficient to yield convergence to a feasible point from $x_{bc}^{(0)}$. Either increasing $\mu^{(0)}$ to 1000 or increasing $\nu^{(0)}$ to 10 was sufficient to ensure convergence to the solution ($\mu^{(0)} = 100$ and $\nu^{(0)} = 1$ was not sufficient). Row W9b lists the results for $\mu^{(0)} = 1000$ and $\nu^{(0)} = 1$, and row W9c gives those for $\mu^{(0)} = 1$ and $\nu^{(0)} = 100$.

Comparison with other algorithms on problems C1, C2, and C3 shows that the algorithm is competitive in practice. The fact that the algorithm took slightly longer than the others reflects the fact that algorithm A can not make unlimited increases in the penalty parameters at each iteration. Using $\mu^{(0)} = \nu^{(0)} = 50$ C1 was solved in 5 iterations with 8 function evaluations. Problem C3F was solved in 3 iterations with 3 function evaluations using $\mu^{(0)} = \nu^{(0)} = 1500$. The results in Table 1 also show the competitiveness of algorithm A.

problem	h_S	j_P	h_P	j_C	h_C	$\theta^\sharp$	$r^\sharp$	$\mu^\sharp$	$\nu^\sharp$
C1	8	6	4	13	7	8.9e-16	3.1e-6	32.24	44.18
C2F	16	17	16	18	18	1.8e-15	3.0e-7	4.74	1
C2I	-	-	-	20	20	0	5.2e-6	5.19	4.83
C3F	3	3	2	7	7	4.9e-11	3.2e-7	1530	1534
C3I	-	-	-	9	9	5.3e-11	5.5e-7	1818	1345
C4	-	-	-	43	27	0	8.8e-7	1	1
C6	-	-	-	95	35	2.3e-13	9.9e-11	96.67	186.5
C7	-	-	-	28	18	4.4e-16	4.0e-7	306.0	75.15
W9a	-	-	-	40	19	7.0e-14	1.3e-6	24.76	18.85
W9b	-	-	-	44	10	5.3e-15	7.2e-6	5778	1
W9c	-	-	-	48	14	1.8e-15	3.8e-7	1660	2542

Table 2: Results for the Colville set of problems and the 'Wright 9' problem.

	SPEPF			HEPF			
problem	j_C	h_C	$\mu^\sharp$	j_C	h_C	$\mu^\sharp$	$\nu^\sharp$
C1	31	11	24.31	13	7	32.24	44.18
C2F	21	20	2.958	18	18	4.74	1
C2I	22	18	3.375	20	20	5.19	4.83
C3F	11	11	1478	7	7	1530	1534
C3I	20	18	1819	9	9	1818	1345
C4	43	27	1	43	27	1	1
C6	220	64	194.6	95	35	96.67	186.5
C7	37	20	336.8	28	18	306.0	75.15

Table 3: Results for the Colville set of problems and the 'Wright 9' problem.

The algorithm's ability to cope with remote starting points was tested using problems C1 and C2. These results are listed in Table 4. Using 10^6 times the initial point, on problem C1 the algorithm took 31 function evaluations and 25 gradient evaluations to solve the problem.

Algorithm A was modified to use the Single Parameter Exact Penalty Function (SPEPF) $f(x)+\mu\theta(x)$ by fixing $\nu = 0$ at every iteration. The results for algorithm A using this SPEPF are listed in Table 3, together with the results for the Hybrid Exact Penalty Function (HEPF). These results show that the HEPF is the better of the two penalty functions.

The nature of the HEPF was explored further using problems P1 and P2, which are listed below.

problem	initial point*100				initial point*10000			
	j_C	h_C	$\mu^\sharp$	$\nu^\sharp$	j_C	h_C	$\mu^\sharp$	$\nu^\sharp$
C1	9	7	80.19	82.96	15	13	4031	20672
C2F	60	46	53.35	1	108	83	38.09	1
C2I	49	40	152.8	1	135	115	15159	1

Table 4: Results for the problems C1 and C2 from remote starting points.

μ	ν	j_C	h_C
1	0	9	8
3	0	14	8
10	0	33	12
30	0	129	28

μ	ν	j_C	h_C
1	1	11	9
1	10	14	8
1	100	30	11
1	1000	74	19

Table 5: Results for problem P2 for various penalty parameter values.

Problem P1:

$$\min_{x_1,x_2} x_2 - e^{x_2-2} \text{ subject to } 1 - x_1^2 - x_2^2 \geq 0, \text{ with } x^{(0)} = (1,8).$$

The solution is $x^* = (0,-1)$. Using the HEPF, the algorithm found this point in 14 iterations and 31 function evaluations, where $\mu^\sharp = \nu^\sharp = 1$; with the SPEPF 60 iterations and 547 function evaluations were performed in solving problem P1, and $\mu^\sharp = 2184164$. This shows a clear superiority of the hybrid (or two parameter) exact penalty function.

Problem P2:

$$\min_{x_1,x_2} x_2^2 - \tfrac{7}{2}x_2 \text{ subject to } 1 - x_1^2 - x_2^2 \geq 0, \text{ with } x^{(0)} = (\tfrac{9}{10},0).$$

The nature of the hybrid exact penalty function was also explored using this problem. Results were generated for a variety of penalty parameter values. On each run μ and ν were fixed. In order to ensure that the NLP's solution is a critical point of the penalty function Φ, values of μ less than 1 were not considered. The results are listed in Table 5 and show that excessive values of either penalty parameter impair the algorithm's performance. However, an excessive ν is much less damaging than an excessive μ. The results for problems C1 and C2 using remote starting points also reflect this.

Problem P2's results suggest that decreasing the penalty parameters should be considered. Sahba [12], and Pantoja and Mayne [6] consider increasing and decreasing the penalty parameter when a single parameter exact penalty function based on the infinity norm is used. Their results show that this can be effective in practice. An advantage to having two penalty parameters is that when μ is decreased a corresponding increase can be made to ν so that the effective penalty parameter value $\mu + \nu\theta$ is decreased only at points at which the maximum constraint violation

θ is small. At points where θ is larger $\mu + \nu\theta$ will increase. This reduces the risk of a decrease in μ permitting a substantial increase in θ. If only a finite number of decreases is permitted then convergence can still be established using theorem 4.1.

6 Concluding Remarks.

The use of a hybrid (or two parameter) penalty function has the additional advantage that, with $\nu > 0$, the QP subproblems are strictly convex; this enlarges the class of QP subroutines capable of solving them.

Badly scaled sets of constraints can be dealt with by scaling each constraint by a positive constant, and adjusting these constants at most a finite number of times throughout the solution process. With this modification the situation is similar to that for an ℓ_1 exact penalty function, where each constraint has a separate penalty parameter, and each such parameter is altered at most a finite number of times.

Exact penalty functions based on the infinity norm have an advantage over one-norm based exact penalty functions in that only the gradients of the most violated constraints need be calculated in order to find a search direction: for ℓ_1 exact penalty functions the gradients of all active and violated constraints may be required.

The algorithm has been shown to generate convergent sequences under mild conditions. Superlinear convergence is obtainable on problems for which f and c are sufficiently continuous [5, 9, 11] when any appropriate update for H is used. The numerical results show that algorithm A is effective in practice, and that use of a second penalty parameter significantly reduces the effort required to solve constrained nonlinear programmes.

References

[1] Bartholomew-Biggs M. C., *Recursive quadratic programming methods based on the augmented Lagrangian*, Math. Prog. Study 31, pp 21–41 (1987).

[2] Fletcher, R, *Practical Methods of optimization*, Second Edition, John Wiley & Sons (1987).

[3] Han, S-P, *A globally convergent method for nonlinear programming*, J. Opt. Th. Appl. 22, 297-309 (1977).

[4] Hock, W. and K. Schittkowski, *Test examples for nonlinear programming codes*, Springer-Verlag, Berlin, ©1981.

[5] Mayne, D. Q. and E. Polak, *A superlinearly convergent algorithm for constrained optimization problems*, Math. Prog. Study 16, 45-61 (1982).

[6] Pantoja, J. F. E. De O., and D. Q. Mayne, *Exact penalty function algorithm with simple updating of the penalty parameter*, J. Opt. Th. Appl. 69, pp 441–467 (1991).

[7] Price, C. J. and I. D. Coope, *An exact penalty function algorithm for semi-infinite programmes*, BIT 30, 723-734 (1990).

[8] Price, C. J., *Nonlinear semi-infinite programming*, PhD thesis, Department of Mathematics, University of Canterbury (1992).

[9] Price, C. J., *Convergence properties of an algorithm for nonlinear programmes which uses a two parameter exact penalty function*, Research Report No. 97, Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand (1993).

[10] Powell, M. J. D., *A fast algorithm for nonlinearly constrained optimization calculations*, in *Numerical Analysis, Dundee 1977* , ed. G.A. Watson, Lecture Notes in Mathematics 630, Springer-Verlag, Berlin (1978).

[11] Powell, M. J. D., *The convergence of variable metric methods for nonlinearly constrained optimization calculations*, in *Nonlinear Programming 3*, eds. O.L. Mangasarian, R.R. Meyer, and S.M. Robinson, Academic Press, New York, (1978).

[12] Sahba, M., *A globally convergent algorithm for nonlinearly constrained optimization problems*, J. Opt. Th. Appl. 52, pp 291–309 (1987).